

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-004261

(43)Date of publication of application : 06.01.1999

(51)Int.Cl.

H04L 12/56  
G06F 13/00

(21)Application number : 10-140964

(71)Applicant : INTERNATL BUSINESS MACH  
CORP <IBM>

(22)Date of filing : 22.05.1998

(72)Inventor : PHILIP SHIRUN YU

(30)Priority

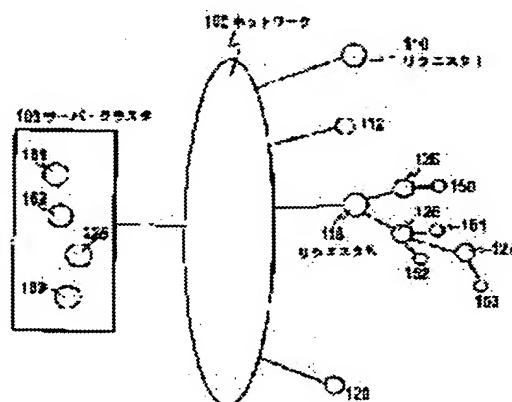
Priority number : 97 866461 Priority date : 30.05.1997 Priority country : US

## (54) DYNAMIC ROUTING METHOD AND PROGRAM STORAGE DEVICE

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To improve either a caching efficiency of a server and a load balance or only the load balance by dynamically routing an object request in a set of server node and optimizing a cache hit rate.

**SOLUTION:** Requesters 110 to 153 request a server cluster 103 for requests by way of a network 105. The server cluster 103 can process requests at any server nodes 161 to 163, but a routing request to a single server of the same object brings improvement of a cache hit rate at the server node. For this purpose, an arbitrator 235 supervises load of each server 161 to 163, dynamically updates a cluster server assignment and improves a load balance. Thus, not only the load balance among server nodes 161 to 163 within the server cluster 103 but also a high cache hit rate are provided.



## LEGAL STATUS

[Date of request for examination] 23.10.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2970760

[Date of registration] 27.08.1999

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-4261

(43) 公開日 平成11年(1999) 1月6日

(51) Int.Cl.<sup>5</sup>  
H 0 4 L 12/56  
G 0 6 F 13/00

識別記号  
3 5 5

F I  
H 0 4 L 11/20  
G 0 6 F 13/00

1 0 2 D  
3 5 5

審査請求 未請求 請求項の数40 O L (全 23 頁)

(21) 出願番号 特願平10-140964  
(22) 出願日 平成10年(1998) 5月22日  
(31) 優先権主張番号 08/866461  
(32) 優先日 1997年5月30日  
(33) 優先権主張国 米国 (US)

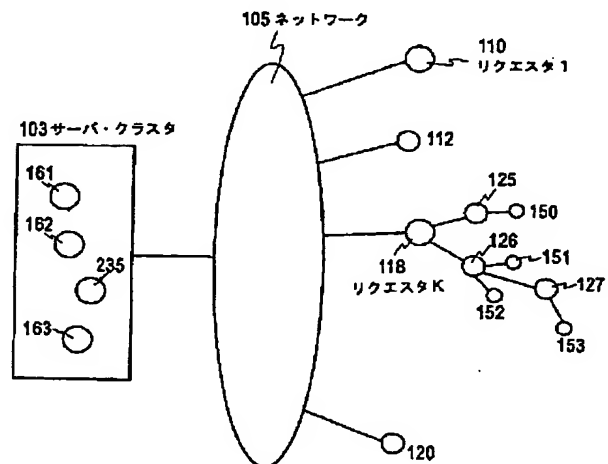
(71) 出願人 390009531  
インターナショナル・ビジネス・マシー  
ズ・コーポレーション  
INTERNATIONAL BUSIN  
ESS MACHINES CORPO  
RATION  
アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)  
(72) 発明者 フィリップ・シールン・ユ  
アメリカ合衆国10514 ニューヨーク州チ  
ャップクァ ストーンウェイ 18  
(74) 代理人 弁理士 坂口 博 (外1名)

(54) 【発明の名称】 動的ルーティング方法及びプログラム記憶装置

(57) 【要約】

【課題】 サーバのキャッシング効率や負荷平衡を向上させるためにサーバの集合またはクラスタでのオブジェクト要求の動的ルーティング方法を提供すること。

【解決手段】 サーバ上のルーティング情報はメタ情報を要求への応答に便乗させて動的に更新することができる。サーバでのキャッシュ・ヒット率を向上させるためサーバ選択が要求オブジェクトの識別子 (例: URL) の因子となる。区画化方法によってオブジェクト識別子をクラスにマッピングできる。また、リクエスト・ノードはサーバ割り当てテーブルを保守して各クラスをサーバ選択にマッピングする。クラス-サーバ割り当てテーブルは作業負荷の変化に伴って動的に変化し、サーバ容量にも影響する。リクエスト・ノードはクラス-サーバ割り当ての動的な変化について「オンデマンド」ベースで通知を受けるだけでよい。



**【特許請求の範囲】**

【請求項 1】オブジェクト要求を処理するサーバ・ノードの集合が用いる動的ルーティング方法であって、メタ情報を要求されたオブジェクトに便乗させるステップと、メタ情報に従ってサーバに対するルーティング情報を動的に更新するステップを含む方法。

【請求項 2】サーバ・ノード間で負荷を平衡させるステップと、

要求されたオブジェクトのキャッシュ・ヒット率を最適化するステップをさらに含む請求項 1 に記載の方法。

【請求項 3】サーバ・ノード間で負荷を平衡させるステップをさらに含む請求項 1 に記載の方法。

【請求項 4】サーバの集合がインターネット環境でのプロキシ・サーバのクラスまたはウェブ・サーバのクラスを含む請求項 1 に記載の方法。

【請求項 5】オブジェクト要求を処理する複数のプロキシ・サーバ・ノード間の動的ルーティング方法であって、

要求されたオブジェクトのオブジェクト識別子に従ってサーバを割り当てるステップと、

前記割り当てステップにตอบสนองして、更新されたサーバ割り当てをオブジェクト・リクエストに通知するステップを含む方法。

【請求項 6】オブジェクト要求を処理する複数のウェブ・サーバ・ノード間の動的ルーティング方法であって、同じホスト名またはアドレスに向けられたオブジェクト要求を、要求されたオブジェクトの識別子に従ってクラス内の異なるサーバに割り当てるステップと、更新されたサーバ割り当てを、オブジェクト・リクエストに通知するステップと、

オブジェクト・リクエストが後続のオブジェクト要求に対する要求されたオブジェクトの更新されたサーバ割り当てを動的に保守するステップを含む方法。

【請求項 7】オブジェクト識別子の階層的マッピングに従ってサーバを割り当てるステップをさらに含む請求項 1 に記載の方法。

【請求項 8】オブジェクト識別子が URL である請求項 7 に記載の方法。

【請求項 9】オブジェクト識別子をクラスにマッピングするステップと、

クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む請求項 2 に記載の方法。

【請求項 10】前記マッピング・ステップがオブジェクト識別子をハッシュ・テーブル経由でクラスまたはハッシュ・クラスにマッピングするステップをさらに含む請求項 9 に記載の方法。

【請求項 11】各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、

クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、

前記マッピング・ステップにตอบสนองしてクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される請求項 2 に記載の方法。

【請求項 12】各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、

クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、

前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される請求項 2 に記載の方法。

【請求項 13】前記負荷平衡ステップが各クラスに関連付けられた負荷の関数としてのクラスをサーバ・ノードに割り当てるステップをさらに含む請求項 2 に記載の方法。

【請求項 14】前記負荷平衡ステップがクラスをサーバに増加方向に再び割り当てるステップをさらに含む請求項 2 に記載の方法。

【請求項 15】前記便乗ステップが PICS プロトコルの使用を含みルーティング情報を更新する請求項 1 に記載の方法。

【請求項 16】各要求元ノードがオブジェクトの現在のサーバ割り当て要求を送信するステップをさらに含む請求項 1 に記載の方法。

【請求項 17】前記要求送信ステップが PICS プロトコルを使って要求されたオブジェクトのクラスに基づいて現在のサーバ割り当てを決定する請求項 16 に記載の方法。

【請求項 18】オブジェクト識別子の階層マッピングに従ってサーバを割り当てるステップが、各クラスを仮想サーバ・ノードに割り当てるステップと、

仮想サーバ・ノードを実際のサーバ・ノードの動的にマッピングするステップをさらに含む請求項 7 に記載の方法。

【請求項 19】サーバ・ノードの集合が、前記の動的マッピング・ステップがネームアドレス・マッピング及びネームアドレス・マッピングに関連付けられたタイムアウト (TTL) 間隔を含み、

DNS が TTL より短い間隔で実際のサーバ・ノードに仮想サーバ・ノードを動的にマッピングするステップと、

更新されたサーバのマッピングをすべてのサーバに通知

するステップをさらに含み、

前記メタ情報が更新されたサーバのマッピングを含み、  
前記の動的にルーティング要求を更新するステップが更新されたサーバ・マッピングに従って後続のオブジェクト要求をルーティングするステップを含むドメイン・ネーム・サーバ（DNS）を含む請求項 18 に記載の方法。

【請求項 20】サーバ・ノードの集合が TCP ルータを含み、ルータが実際のプロキシ・ノードに仮想サーバ・ノードを動的にマッピングするステップをさらに含む請求項 18 に記載の方法。

【請求項 21】すべてのリクエストが前記の動的更新ステップを実行するように適合されていない異種リクエスト環境をさらに含む請求項 1 に記載の方法。

【請求項 22】前記サーバ割り当てステップがオブジェクト識別子の階層マッピングに従ってサーバを割り当てるステップを含む請求項 5 に記載の方法。

【請求項 23】オブジェクト識別子をクラスにマッピングするステップと、

クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む請求項 5 に記載の方法。

【請求項 24】各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、

クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、

前記マッピング・ステップにตอบสนองしてクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される請求項 5 に記載の方法。

【請求項 25】各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、

クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、

前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される請求項 5 に記載の方法。

【請求項 26】前記オブジェクト要求割り当てステップがオブジェクト識別子の階層マッピングに従ってオブジェクト要求を割り当てるステップを含む請求項 6 に記載の方法。

【請求項 27】オブジェクト識別子をクラスにマッピングするステップと、

クラス及びクラスサーバ割り当てテーブルに基づいて

サーバを割り当てるステップをさらに含む請求項 6 に記載の方法。

【請求項 28】各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、

クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、

前記マッピング・ステップにตอบสนองしてクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される請求項 6 に記載の方法。

【請求項 29】各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、

クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、

前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される請求項 6 に記載の方法。

【請求項 30】サーバ・ノードの集合への要求をクラス内の異なるサーバに割り当てることができるサーバ・ノードの集合の動的なルーティング方法であって、

リクエストがリクエスト識別子または IP アドレスのいずれかを含むマッピング要求を定期的にサーバに送信するステップと、

前記のリクエスト識別子または IP アドレスのいずれかをリクエスト負荷及びサーバ容量のいずれかに基づいてサーバ・ノードの集合内のサーバにマッピングするステップと、

前記マッピング・ステップにตอบสนองしてサーバ・マッピングをすべてのサーバに送信するステップと、

サーバのいずれかがそのサーバに割り当てられていないリクエストから要求を受信した場合に、サーバがリクエストにリクエストサーバ割り当ての変更を通知するステップを含む方法。

【請求項 31】リクエストに通知する前記ステップが、サーバが要求を処理するステップをさらに含む請求項 30 に記載の方法。

【請求項 32】リクエスト識別子または IP アドレスをクラスに区画化するステップと、

アービトラータ・サーバとサーバの集合にあるクラスサーバ割り当てテーブルを保守するステップをさらに含む請求項 30 に記載の方法。

【請求項 33】アービトラータ・サーバがインターネット環境の DNS を含む請求項 32 に記載の方法。

【請求項 34】オブジェクト識別子または IP アドレス

の階層マッピングに従ってサーバを割り当てるステップをさらに含む請求項3に記載の方法。

【請求項35】リクエスト識別子をクラスにマッピングし、クラスに基づいてサーバを割り当てるステップをさらに含む請求項3に記載の方法。

【請求項36】サーバが更新されたメタ情報をリクエストへ送信し、リクエストが割り当てを更新するステップをさらに含む請求項3に記載の方法。

【請求項37】マシン上で実行され、オブジェクト要求を処理するサーバ・ノードの集合でオブジェクト要求を動的にルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、メタ情報を要求されたオブジェクトに便乗させるステップと、メタ情報に従ってサーバ割り当てに対するルーティング情報を動的に更新するステップを含むプログラム記憶装置。

【請求項38】マシン上で実行され、オブジェクト要求を処理するサーバ・ノードの集合でオブジェクト要求を動的にルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、要求されたオブジェクトのオブジェクト識別子に従ってサーバを割り当てるステップと、前記割り当てステップにตอบสนองして、更新されたサーバ割り当てをオブジェクト・リクエストに通知するステップを含むプログラム記憶装置。

【請求項39】マシン上で実行され、オブジェクト要求を処理する複数のウェブ・サーバ・ノードでオブジェクト要求を動的にルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、同じホスト名またはアドレスに向けられたオブジェクト要求を、要求されたオブジェクトの識別子に従ってクラスタ内の異なるサーバに割り当てるステップと、更新されたサーバ割り当てを、オブジェクト・リクエストに通知するステップと、オブジェクト・リクエストが後続のオブジェクト要求に対する要求されたオブジェクトの更新されたサーバ割り当てを動的に保守するステップを含むプログラム記憶装置。

【請求項40】マシン上で実行され、オブジェクト要求を処理するサーバ・ノードの集合でオブジェクト要求を動的にルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、リクエストがリクエスト識別子またはIPアドレスのいずれかを包含するマッピング要求を定期的にサーバに送信するステップと、

前記のリクエスト識別子またはIPアドレスのいずれかをリクエスト負荷及びサーバ容量のいずれかに基づいてサーバ・ノードの集合内のサーバにマッピングするステップと、

前記マッピング・ステップにตอบสนองしてサーバ・マッピングをすべてのサーバに送信するステップと、サーバのいずれかがそのサーバに割り当てられていないリクエストから要求を受信した場合に、サーバがリクエストにリクエストサーバ割り当ての変更を通知するステップを含むプログラム記憶装置。

【発明の詳細な説明】

【0001】本発明は次の同時係属の米国特許出願に関連する。1997年7月2日出願のP. ユー (Yu) の「A Method and Apparatus for Dynamic Interval-based Balancing」という名称の米国特許出願第08/798385号 (IBM整理番号Y0997028) 及び1996年5月12日 (仮) 出願のディアス (Dias) 他「A Computer System and Method for Load Balancing with Selective Control」という名称の米国特許出願 (暫定) 第60/031849号 (IBM整理番号Y0996-252) に関連する。これらの同時出願及び本発明は共に米国ニューヨーク州Armonkのインターナショナル・ビジネス・マシーンズ・コーポレーションを譲受人としている。これらの同時係属の出願に記載された内容はそのまま本明細書に参照として組み込まれている。

【0002】

【発明の属する技術分野】本発明は一般にインターネット環境でのプロキシ・サーバやウェブ・サーバなどのサーバの集合体またはクラスタ間での負荷の均衡をとることに関する。本発明の詳細な態様はクライアント要求への応答に便乗したメタ・データを用いたルーティング情報の更新方法に関する。さらに別の態様はキャッシング効率を最適化する負荷の均衡をとる方法に関する。

【0003】用語集

本明細書で使用する用語の一部は辞書にある意味もあるが、次の用語集も役立つであろう。

【0004】インターネット

一連のTCP/IPプロトコルを使用するネットワーク及びゲートウェイからなるネットワーク。

【0005】クライアント

クライアントとは、コマンドをサーバに発行してそのコマンドに対応するタスクを実行させるコンピュータである。

【0006】サーバ

他のコンピュータのコマンドに従ってタスクを実行する任意のコンピュータがサーバである。ウェブ・サーバは通常1つまたは複数のクライアントをサポートする。

【0007】ワールド・ワイド・ウェブ（WWWまたはウェブ）

強調表示した言葉や関心がある語句（ハイパーリンク）をクリックしてサーバ相互間及びデータベース相互間でインターネット・スイッチ上で情報を検索するためのインターネットのアプリケーション。インターネットWWWサーバはクライアントをサポートし、情報を提供する。ウェブはすべての資源がURLとしてアドレス指定され、HTMLを使ってURLに対応する情報を表示し、他のURLへポイントアンドクリックによるインタフェースを提供するインターネットと考えられる。

【0008】ユニバーサル・リソース・ロケータ（URL）

インターネット上の情報を一意的に識別する、またはアドレス指定する方法。Eメール・アドレスのウェブ・ドキュメント・バージョンまたは完全に資格があるネットワーク・ファイル名。これらはハイパーリンクでアクセス可能である。URLの1例は、「http://www.philipyu.com:80/table.html」である。ここでURLは4つのコンポーネントを備える。左から始めて、最初は残りのロケータと“:”で区別される使用プロトコルで規定する。2番目は対象ホストのホスト名またはIPアドレスである。これは左側が“/”で、右側が“/”で区切られ、任意選択として“:”で区切られる。ポート番号は任意選択で、左側はホスト名“:”で区切られ、右側は“/”で区切られる。4番目のコンポーネントは実際のファイル名またはプログラム名である。この例では、「.html」の拡張子はHTMLファイルであることを示す。

【0009】ハイパーテキスト・マークアップ言語（HTML）

HTMLはとりわけウェブ・サーバが使用してドキュメントを作成し、ウェブ・クライアントから表示できる他のウェブ・ドキュメントへ接続するための言語である。HTMLはハイパーテキスト・ドキュメントを使用する。

【0010】ハイパーテキスト転送プロトコル（HTTP）

HTTPはクライアントからサーバへのすべての要求が独立して扱われるということの意味するステートレス・プロトコルの例である。サーバは以前の接続の記録を持たない。URLの最初の「http:」はhttpを使ってファイルを検索する必要があることを示す。

【0011】インターネット・ブラウザまたはウェブ・ブラウザ

httpなどのインターネット・プロトコルを実行してその実行結果をユーザの画面に表示するグラフィカル・インタフェース。ブラウザはユーザがインターネットの「サーフィン」をする際のデスクトップ画面、ディレク

トリ及び検索ツールを備えたインターネットのツアー・ガイドとして機能できる。この適用分野ではウェブ・ブラウザはワールド・ワイド・ウェブと通信するクライアント・サービスである。

【0012】クライアント・キャッシュ

クライアント・キャッシュは通常クライアントがアクセスするオブジェクトの1次キャッシュとして使用される。WWW環境では、クライアント・キャッシュは通常ウェブ・ブラウザによって実施され、現在の起動の間にオブジェクトをキャッシングする、すなわち、非持続性キャッシュであるか、複数の起動にまたがってオブジェクトをキャッシングできる。

【0013】キャッシング・プロキシ

クライアントのためにエージェントとして動作し、オブジェクトのキャッシングされたコピーを見つけるネットワーク内の専用サーバ。キャッシング・プロキシは、クライアント・キャッシュからのキャッシュ欠落の結果として起動されるため、通常は2次またはそれ以上のレベルのキャッシュとして動作する。

【0014】HTTPデーモン（HTTPD）

ハイパーテキスト・マークアップ言語及び共通ゲートウェイ・インタフェース機能を備えたサーバ。HTTPDは通常、イントラネットに上のマシンへのハードウェア接続とTCP/IPカプリングなどのインターネットへのアクセスを提供するアクセス・エージェントによってサポートされる。

【0015】

【従来の技術】ワールド・ワイド・ウェブ（WWW）上のトラフィックは指数関数的に増加している。大規模な組織または領域へのゲートウェイにあるプロキシ・サーバはコンピュータ・ノードの集合を含むことができる。同様に、人気が高い（ホットな）Webサイトでは、コンピュータ・ノードの集合（またはクラスタ）がアクセス需要をサポートするために使用される。

【0016】サーバ・クラスタを高パフォーマンスに保つには、ノードの集合間で負荷を分散する必要がある。これは同一のオブジェクト要求をローカライズする方法でクラスタ内の任意のサーバのキャッシュ・ヒット率を最適化する必要性によって軽減される。

【0017】IBM S/390 Sysplexなどのマルチプロセッサまたは複数のノード環境での負荷平衡の以前の処理は主にインカミング・タスクまたはユーザ・セッションごとに複数の汎用資源の1つを選択するスケジューリング・アルゴリズムに焦点を合わせていた。スケジューラはすべてのインカミング・タスクまたはユーザ・セッションのスケジューリングを制御するため、資源選択のキャッシングは行われなかった。

【0018】地理的に分散した複製サイト間に負荷平衡を行う周知の方法の1つは、ラウンドロビン・ドメイン・ネーム・サーバ（RR-DNS）アプローチとして知

られる。Computer Network and ISDN Systems、第27巻、1994年、pp. 68-74所収のカッツ (Katz.)、E、バトラー (Butler)、M.、及びマグラス (McGrath)、R. の論文「A Scaleable HTTP Server: The NCSA Prototype」では、ウェブ・サーバ・ノードのセット全体でノードの負荷平衡を行う方法が紹介されている。ここで、分散サイトのセットがURL (例えばwww. hotsite. com) によって表されている。この分散サイトのクラスタ・サブドメインはそのサブドメイン・ネーム・サーバで定義される。サブドメイン・ネーム・サーバは名前解決要求を異なるIPアドレス (分散クラスタ内の) ヘラウンドロビン方式でマッピングする。このように、クライアントのサブセットは複製サイトのそれぞれに割り当てられる。ネットワーク・トラフィックを軽減するために、マッピング要求はサーバ要求ごとには発行されない。その代わりにマッピング要求の結果が「タイムトゥリブ」(TTL) 間隔だけ保存される。TTL間隔内に発行されたその後の要求は以前のマッピングを保持しているので同じサーバ・ノードヘルレーティングされる。

【0019】RR-DNS方式の問題は、分散サイト間の負荷の不均衡が招来される可能性があることである (例えば、ディアス (Dias)、D. M.、キッシュ (Kish)、W.、マッカージー (Mukherjee) R.、及びテワリ (Tewari)、R. の「A Scaleable and Highly Available Web Server」、Proc. 41st IEEE Computer Society Intl. Conf. (COMPCON) 1996、Technologies for the Information Superhighway, pp. 85-92、1996年2月を参照)。負荷の不均衡はネットワーク内のさまざまなゲートウェイ、ファイアウォール、及びドメイン・ネーム・サーバでの名前とIPアドレス間の対応のキャッシングによって引き起こされる場合がある。このように、TTL間隔の間はこれらのゲートウェイ、ファイアウォール、及びドメイン・ネーム・サーバを介してルーティングされる新しいクライアント要求はすべてキャッシュに記憶された単一のサイトに割り当てられる。当業者はTTL値を減らすことだけでは問題の解決にならないことを理解するであろう。実際、多くのネーム・サーバは小さいTTLの値を受け付けない場合が多い。さらに重要なこととして、TTLの値を小さくしても不均衡に分散されたクライアント要求率に起因する負荷の偏りは是正されない可能性がある。

【0020】ローカル・クラスタまたはノードでの負荷平衡の1つの方法は、アタナシオ (Attanasio)、クレメント (Clement) R. 及びスミス

(Smith)、ステファン (Stephen) E.、  
「A Virtual Multi-processor Implemented by an Encapsulated Cluster of Loosely Coupled Computers」、IBM Research Report RC 18442、1992年及び参照により本明細書にそのまま組み込まれる「Method and Apparatus for Making a Cluster of Computers Appear as a Single Host」1994年12月6日の米国特許第5371852号に記載されたいわゆるTCPルータを使用する方法である。ここでは、TCPルータのアドレスだけがクライアントに提示されている。TCPルータは着信要求をクラスタのノード間にラウンドロビン法またはノード上の負荷に基づいて分散する。このTCPルータ方法はノードのローカル・クラスタに限定されることに注意されたい。

【0021】さらに最近になって、本明細書に参照によりそのまま組み込まれているコエアジャンニ (Cola janni)、M.、ユー (Yu)、P.、及びディアス (Dias)、D. の論文「Scheduling Algorithms for Distributed Web Servers」、IBM Research Report, RC20680、1997年1月では、ゲートウェイをその要求率に従って複数の層に分割する複数層ラウンドロビン方式が記述されている。各層からの要求はラウンドロビン・アルゴリズムを使って別々にスケジューリングされる。この方式は同種の分散サーバ環境を処理することもできる。

【0022】上記の方法のいずれにおいても、目標はサーバの集合の間で負荷を分散することである。動的ルーティング決定は、要求されているオブジェクトの識別を考慮しない。換言すると、同じオブジェクトへの複数の要求は別々のサーバヘルレーティングされて負荷が分散される。この方法によって、地域のウェブ・ページの潜在的な数は膨大になり得るためにプロキシ・サーバでは特に高いキャッシュ・ヒット率を抑えることができる。ウェブ・サーバ・クラスタ内ではそれぞれの区画に異なる (仮想) ホスト名またはIPを割り当てる静的区画がウェブ・ページに対して行われるが、この静的区画方式は動的な負荷の変更に柔軟に対応できず、さらにスケーラブルでもない。

【0023】以上のように、改良された負荷平衡方法及び同一のオブジェクト要求をローカライズすることでクラスタ内で負荷平衡を行うだけでなくクラスタ内の任意のサーバのヒット率を最適化するサーバ・クラスタ内の装置が必要とされる。本発明はこのような必要性に対処する。

【0024】さらに、作業負荷条件に従って各サーバに



オブジェクト・スペースのサブセットを動的に処理する方法を割り当て、オブジェクトに関連付けられたサブスペースに割り当てられたサーバへオブジェクト要求をルーティングする改良されたルーティング方法も必要とされる。本発明はこのような必要性にも対処する。

【0025】

【発明が解決しようとする課題】上記の必要性に従って、本発明の目的は、サーバのキャッシング効率及び負荷平衡、または負荷平衡のみのいずれかを考慮するサーバの集合でのオブジェクト要求の動的なルーティングを行う改良された方法及び装置である。

【0026】本発明はルーティング要求への応答にメタ情報を「便乗させる」ことでサーバ・ルーティング情報を動的に更新できる機能をさらに備える。本発明はまた要求オブジェクトの識別（例えばURL）に基づいてサーバをマッピングし、作業負荷条件が変更された場合にこのマッピングを動的に更新することでサーバでのキャッシュ・ヒット率を改善できる他の機能も備える。インターネット環境では、サーバの集合はこれに限定されないが、プロキシ・サーバ・クラスタまたはウェブ・サーバ・クラスタを含むことができる。

【0027】サーバ・ノードの集合でオブジェクト要求を動的にルーティングする本発明の機能を備えた方法は、メタ情報を要求オブジェクトに便乗させるステップと、メタ情報に従ってサーバ割り当てを行うためのルーティング情報を動的に更新するステップを含む。

【0028】

【課題を解決するための手段】サーバ・ノードの集合でオブジェクト要求を動的にルーティングしながらキャッシュ・ヒット率を最適化する本発明の機能を備えた方法は、オブジェクト識別をクラスにマッピングし、クラス及びクラスサーバ割り当てテーブルに従ってサーバを割り当てるステップをさらに含む。

【0029】本発明は「オンデマンド」ベースでリクエスト・ノードにクラスサーバ割り当ての動的な変更を通知できる他の機能をさらに含む。クラスサーバ割り当ては作業負荷の変化に応じて動的に変更できる。この変更をすべての潜在的なリクエストへ同報通信するコストを避けるため、またはリクエストに要求が送信されるたびにまずマッピングを取得させるために、サーバはそのクラスの処理を割り当てられていないオブジェクト要求の処理を継続できるという点で有利である。ただし、サーバは返送されたオブジェクト（または応答）のヘッダで新しいクラスサーバ割り当てに関する情報を示すことができる。

【0030】さらに、メタ情報を要求オブジェクトに便乗させる本発明の機能はインターネットの従来のDNSルーティングにも応用でき、これによってサーバ・クラスタの負荷平衡を向上させられる。これはサーバ割り当て（キャッシュ・ヒット改良のための）のためにオブジ

ェクトのURL（またはオブジェクト・クラス）を使用するという概念と区別しなくてはならない。DNSルーティングはアドレス・マッピングの有効間隔（TTL）を備える。本発明はサーバ割り当てをTTLより小さい間隔で生成し、真の負荷条件をよりよく反映するための機能を備える。サーバ割り当ての変更を返送されたオブジェクトに便乗させてトラフィックの追加を回避し、将来の要求を新しいサーバへ送信できる。

【0031】本発明は、負荷の均衡をとるためクラスサーバ割り当てを作業負荷需要に基づいて動的かつ増分的に変更できる他の特徴をさらに有する。

【0032】本発明のさらに別の機能によれば、インターネット環境でPICSプロトコルを使ってさまざまなタイプの情報の通信を行うことができる。PICSを使ってサーバは要求が廃止されたクラスサーバ・マッピング・エントリに基づいて新しいクラスサーバ・マッピングにメタ情報を便乗できる。またPICSを使ってリクエストはコーディネータに現在のクラスサーバ・マッピングを照会することができる。

【0033】当業者は本発明がワールド・ワイド・ウェブだけでなく一般の分散環境に適用できることを理解するであろう。

【0034】

【発明の実施の形態】図1は本発明に適用可能なインターネット環境の図である。図示のように、リクエスト（110～153）は従来のプロキシ・サーバ・ノード（118、125～127）と、クライアント・ワークステーション及びパーソナル・コンピュータ（PC）（110、112、120、150～153）のすべてを含むことができ、ネットワーク（105）に接続される。プロキシ・サーバ、ワークステーション、及びPCは当業では周知である。プロキシ・サーバの例はIBMが販売するインターネット接続サーバ（ICS）である。リクエストはサービス・クラスタ（103）にネットワーク（105）経由でサービスを要求する。ネットワークの例は、これに限定されないが、インターネット、ワールド・ワイド・ウェブ（WWW）、イントラネット、及びローカル・エリア・ネットワーク（LAN）を含む。サーバ・クラスタは高トラフィック需要を処理する複数のサーバ・ノード（161～163）を含む。サーバ・クラスタはプロキシ・サーバまたはウェブ・サーバ・クラスタでよい。クラスタ内のサーバは、これに限定されないが、IBMがS/390 SYSPLEX、SP2、またはRS6000ワークステーションの商標名で販売する製品を含む。従来と同様、各要求はクラスタ内のすべてのサーバが処理できる。一般のサービス要求はワールド・ワイド・ウェブ（WWW）ページへのアクセス、リモート・ファイル転送、電子メール、及びトランザクション・サポートを含む。

【0035】要求は原則としてどのサーバ・ノードでも

処理できるが、同じオブジェクトの単一サーバ・ノードへのルーティング要求はそのサーバ・ノードでのキャッシュ・ヒット率の向上を、したがって性能の改善をもたらす。以下に説明するように、本発明はクラスタ内のサーバ・ノード間の負荷平衡だけでなく高いキャッシュ・ヒット率を実現する機能を備える。

【0036】概要によって、本発明によるルーティング方法はオブジェクトの論理識別またはシンボル名（例：URL）を使って、要求を処理するサーバを選択する。オブジェクト識別子をクラスにマッピングする区画方法も提供される。また好ましくは要求ノードが各クラスを選択したサーバに割り当てするためのクラスサーバ割り当てテーブル（図3）を保守する。好ましい区画方法は従来のハッシュ関数を使ってオブジェクトURLを任意の数のハッシュ・クラスにハッシュする方法である。ハッシュ関数が好ましくは与えられ、アービトラータ235（図2）によってすべての参加サーバ及びリクエスト・ノードから認識できる。

【0037】アービトラータ235は各サーバの負荷を監視し、クラスサーバ割り当てを動的に更新して負荷平衡を向上させる。本発明はサーバ103によってクラスサーバ割り当てが動的に変更された場合にリクエスト・ノードに「オンデマンド」で通知する方法も提供する。

【0038】リクエスト・ノードからの要求はサーバ・クラスタ103に到達する前にいくつかの中間リクエスト・ノード（すなわち、プロキシ・サーバ）を経由する必要がある。例えばノード150はサーバ・クラスタ103に到達する前に2つのレベルのプロキシ・ノード125及び118を経由する必要がある。サーバ・クラスタがプロキシ・サーバ・クラスタの場合、サーバの選択はプロキシ・サーバ・クラスタ103に最も近いリクエスト110～120によって行われるのが好ましい。ウェブ・サーバ・クラスタの場合、ウェブ・サーバの選択はパス上の中間リクエストで実行できる。

【0039】本発明はリクエストとサーバ・ノード間で「便乗された」メタデータを使ってルーティング情報を効率的に送受信する機能をさらに備える。HTTPの実施形態では情報交換は既存のウェブ・プロトコルを使ってオブジェクト・ヘッダ内にメタデータとして含むことができる。PICS（「インターネット・コンテンツ用プラットフォーム」）は電子コンテンツに関するメタ情報を送信する方法を規定する。PICSはウェブ・コンソーシアム・プロトコル勧告（<http://www.w3.org/PICS>を参照）である。PICSは「このコンテンツにはどの程度のヌーディティが関連付けられているか。」などの値に基づいた評価ラベルを送信するために初めて使用されたが、メタ情報のフォーマット及び意味は完全に一般的である。PICSでは、電子コンテンツに関するメタ情報は、情報の「評価サービ

ス」またはproducer-and-intended-usage（プロデューサアンドインテンドユーセージ）に従ってグループ化され、そのグループの1つで任意の数のカテゴリまたは大きさを持つ情報が送信できる。各カテゴリはある範囲の使用できる値を備え、特定のコンテンツに関して特定のカテゴリが単一の値または複数の値を備えることができる。さらに、メタ情報グループ（「PICSラベル」として知られる）は満了情報を含むことができる。また、PICSラベルが複数を電子コンテンツに付与する機能もある。特定の電子コンテンツの各PICSラベルは独立してコンテンツに追加または削除できる。

【0040】例えば、「評価サービス」フィールドが「SafeSurf」評価システムによる値ベースの評価ラベルを含むことを示す単一のPICSラベルを備えたサーバからイメージ・ファイルを送信できる。本発明によれば、イメージ・ファイルは企業プロキシを通過する際にその「評価サービス」フィールドがクラスサーバ割り当て情報を含むことを示す第2のPICSラベルを受信することができる。第2のPICSラベルは部門プロキシを通過する際にはがすことができる。このように、クライアント・コンピュータは第1のPICSラベルだけを参照することができる。HTTPプロトコルはPICSをサポートする要求ヘッダ及び応答ヘッダで増強されている。NNTPなどの他の共通アプリケーション・プロトコルを定義する専門機関は現在PICSサポートの追加を検討している。これらのプロトコルの一部として、必要なPICSラベル・タイプのリストを要求を付けて含ませることができる。PICSはまた中央ラベル局サーバからPICS情報を受信するための照会フォーマットを規定する。PICSラベルの例は（PICS-1.1「`http://the.rating.service`」label for「`http://the.content`」exp「1997.07.01T08:15-1500」r(n4s3v210)）で「n」「s」「v」「l」は各種メタ情報タイプの送信名でこのコンテンツの適用可能な値は4（nの場合）、3（sの場合）、2（vの場合）及び0（lの場合）である。ID「`http://the.rating.service`」を認識するソフトウェアだけがこれらのカテゴリ及び値の解釈方法を認識できる。

【0041】好ましい実施形態では、2種類の異なるPICSラベルが使用される。最初の種類のPICSラベルは「再割り当て」ラベルまたは（Rラベル）と呼ばれ、クラスタ内のサーバ・ノードが返送されたオブジェクトのオブジェクト・クラスの「現在の」サーバ割り当てを示すのに用いる。第2の種類のPICSラベルは「割り当て」ラベルまたは（Aラベル）と呼ばれ、リクエストがこの場合にラベル局機能を提供するアービトラータからのオブジェクトのURLの現在のサーバ割り当

てを決定するために用いる。

【0042】図2に本発明の機能を備えたネットワーク(201)及びシステムの例を詳述する。図示のように、リクエスト・ノード(202~203)を使ってネットワーク(201)を介して要求を発行できるコンピュータ・ノードが表される。リクエスト・ノードは好ましくはCPU(260)、RAMなどのメモリ(263)、及びDASDまたはディスクなどの記憶装置(265)、またはその他の安定化磁気、電気、または光記憶あるいはこれらのすべてを含む。メモリ(263)は、好ましくは安定化プログラム記憶(265)からメモリ(263)へロードされてCPU(260)で実行されるコンピュータ実行可能コードとして実施される、本発明によるリクエスト203論理(詳細は図13を参照して後述)を含む。当業者はリクエスト(203)論理をネットワーク(201)経由でリクエストヘダウンドロードしてCPU(260)で実行できることを理解するであろう。リクエスト203論理はオブジェクト要求生成ルーチン(267)(詳細は図14に図示)を含み、クラスサーバ割り当てテーブル(270)のコピーを保守する。

【0043】アービトレータ(235)はサーバ・トラフィックを監視できるすべての従来のコンピュータ・ノードを表し、「クラスサーバ」割り当てを調整できる。アービトレータ(235)は好ましくはCPU(240)、RAMなどのメモリ(245)、及びDASDなどの記憶装置(242)、またはその他の安定化磁気、電気、または光記憶あるいはこれらのすべてを含む。メモリ(245)は、好ましくはプログラム記憶(242)からメモリ(245)へロードされてCPU(240)で実行されるコンピュータ実行可能コードとして実施される、本発明によるアービトレータ論理(詳細は図9で図示)を含む。アービトレータ論理はわかりやすくするために図の例でのみ、マッピング要求ハンドラ(248)と、統計及び評価ルーチン(250)を含むいくつかのコンポーネントに分割されている。これらのコンポーネントは図12及び10を参照してそれぞれ詳述する。保守される主要なデータ構造はクラスブロキシ割り当てテーブル(225)である。クラスブロキシ割り当てテーブル(225)上の動作を各種コンポーネントに関して説明する。

【0044】サーバ1...M(206~208)はデータ/オブジェクトへのアクセスまたはリクエスト(203)の要求によるファイル転送あるいはその両方の提供などのサービス要求を処理することができるすべての従来のコンピュータ・ノードを含むことができる。サーバ・ノード(208)はCPU(227)、メモリ(210)、及びDASDなどの記憶装置(230)、またはその他の安定化磁気、電気、または光記憶装置あるいはそれらの組合せを含む。メモリ(210)は、好まし

くは記憶装置(230)からメモリ(210)へロードされてCPU(227)で実行されるコンピュータ実行可能コードとして実施される、本発明によるサーバ論理(詳細は図4で図示)を含む。サーバ・ノード論理はわかりやすくするために図の例でのみ、オブジェクト要求ハンドラ(212)と、オブジェクト・ハンドラ(214)と、統計報告ルーチン(218)を含むいくつかのコンポーネントに分割されている。これらのコンポーネントはそれぞれ図7、5、及び8でそれぞれ詳述する。また、キャッシュ・マネジャ(220)を含み、クラスサーバ割り当てテーブル(225)のコピーを維持する。

【0045】図3に $N=16$ 及び $M=3$ 、ただし $N$ は好ましくはオブジェクト・クラスの数、すなわち、ハッシュまたは割り当てテーブルのサイズ、 $M$ はサーバ数である割り当てテーブル(225、270)の例を示す。 $C(.)$ をクラス $k$ をサーバ $C(k)$ に割り当てる割り当てテーブル(225、270)とする。再度図2を参照すると、アービトレータ(235)及びクラス内の各サーバ(206、208)ノードだけでなくリクエスト・ノード(202、203)も割り当てテーブル(225、270)のコピーを保守できる。リクエストのテーブル(270)は一般に最新のものではない、すなわち、サーバ(208)またはアービトレータ(235)割り当てテーブル(225)と同期をとっていない。本発明は各テーブルの同期を確保するためにコストがかかる更新メッセージを送信する必要がない機能を備え、好ましくは便乗させられたメタ・データを使って「オンデマンド」でクラスサーバ・マッピングを更新する。

【0046】図4に本発明によってCPU(227)上で実行されるメモリ(210)に記憶されたサーバ(208)論理の例を示す。 $C(.)$ をクラス $k$ をサーバ $C(k)$ に割り当てる割り当てテーブル(225、270)とする。図示のように、ステップ410でサーバは入力を待つ。ステップ415で、受信した入力によって異なる処置がとられる。受信した入力オブジェクト要求の場合、ステップ420でオブジェクト要求ハンドラ212が起動される。オブジェクト要求ハンドラの詳細な例は図7を参照しながら記述する。ステップ430で、受信した入力オブジェクトの場合、ステップ440でオブジェクト・ハンドラ214が起動される。オブジェクト・ハンドラの詳細な例は図5を参照しながら記述する。ステップ445で、受信した入力統計収集要求(アービトレータからの)の場合、ステップ460で統計報告ルーチン(218)が起動される。統計報告ルーチンの詳細な例は図8を参照しながら記述する。ステップ450で、受信した入力割り当てテーブル更新要求(アービトレータからの)の場合、 $C(k)$ 、 $k=1, \dots, M$ がステップ465でこれに従って更新される。本発明では扱わない他のタイプの入力の場合(従

来のHTTP「プル」要求またはFTP要求などの)、該当する各種ハンドラ(470)が起動できる。

【0047】図5にオブジェクト・ハンドラ(214)の例を示す。ステップ510で、受信したオブジェクトのオブジェクト・クラス(図7のステップ750を参照)が割り当てテーブルに示されるようにこのサーバに割り当てられたクラスに属する場合、ステップ515でキャッシュ・マネジャ220が起動される。キャッシュ・マネジャはこのオブジェクトをキャッシングするかどうかを決定し、キャッシングする場合は現在キャッシングされているどのオブジェクトを交換するかを決定する。次にステップ530で、オブジェクトはリクエストへ返送される。ステップ510で、受信したオブジェクトのオブジェクト・クラスがこのサーバに割り当てられたクラスに属していない(割り当てテーブルからわかるように)場合、ステップ520で動的再割り当てルーチンが起動される。動的再割り当てルーチンの詳細な例は図6を参照しながら記述する。

【0048】図6に動的再割り当てルーチン(ステップ520)の例を示す。図示のように、ステップ610で、オブジェクト・クラスを処理するための該当するサーバid(またはIPアドレス)が割り当てテーブル(図3)から決定される。ステップ620で、カテゴリ値がオブジェクトのこのクラス(図3)を処理するために割り当てられるサーバ(図3)を示すラベルがオブジェクト・ヘッダに挿入される。

【0049】図7にオブジェクト要求ハンドラ212の例を示す。図示のように、ステップ710で、オブジェクトがローカル・バッファ内にある場合、割り当てテーブル(225)がチェックされてオブジェクト・クラス(図3)をこのサーバで処理するかがステップ720で決定される。処理しない場合、動的再割り当てルーチンが起動される(図6)。ステップ740で、オブジェクトはリクエストへ返送される。ステップ710で、オブジェクトがローカル・キャッシュにないと判明した場合、ステップ750でオブジェクト取得要求が(リクエストのために)送信される。

【0050】以下の説明では、 $CS(j, i)$ をサーバjが受信した(現在の測定間隔中に)クラスiのオブジェクト要求数とし、 $CA(i)$ をすべてのサーバが受信したクラスi内のオブジェクトに対する要求総数とする。さらに、 $SA(j)$ をサーバjに割り当てられたオブジェクト・クラス要求数とする。

【0051】図8に統計報告ルーチン(218)の例を示す。図示のように、ステップ810で、サーバjはそのロード情報 $CS(i, j)$ (ただし $i=1, \dots, N$ )をアービトラータへ送信する。ステップ820で、 $CS(i, j)$ (ただし $i=1, \dots, N$ )はゼロにリセットされる、すなわち、リセットされて新しい収集または測定間隔のカウントが開始される。

【0052】図9にアービトラータ論理(235)の例を示す。ステップ910で、アービトラータは入力待ちとなる。ステップ920で、マッピング要求が検出されると、マッピング要求ハンドラ(248)がステップ940で起動される(マッピング要求ハンドラ(248)の詳細な例は図12を参照しながら説明する)。ステップ930で、統計収集間隔のタイマの満了が検出されると、アービトラータはステップ950で統計及び評価ルーチン(250)を実行する(統計及び評価ルーチン(250)の詳細な例は図10を参照しながら説明する)。ステップ960で、更新要求が更新割り当てテーブルを備えたすべてのサーバへ送信される。

【0053】図10に統計及び評価ルーチン(250)の例を示す。図示のように、ステップ1010で、統計収集要求がすべてのサーバへ送信されてサーバj、ただし $j=1, \dots, M$ から $CS(j, i)$ 、ただし $i=1, \dots, M$ の値が計算される。ステップ1020で、 $CA(i)$ がクラスごとに計算される(各クラスiの全サーバでの要求総数)。ステップ1030で、 $SA(j)$ が各サーバjについて計算される(各サーバjの割り当てられたクラスへの要求総数)。ステップ1040で、サーバ上の負荷の上限値THが計算される。THは平均負荷より高い分画(d)であると定義することが好ましい。例えば、dが0.2とは負荷平衡の目標が平均値の20%を超えるサーバが全くないことを意味する。ステップ1050で、いずれかのサーバの負荷が上限値THを超えると、再割り当てルーチンが起動されてクラス-サーバ割り当てが調整され、負荷平衡が向上する。再割り当てルーチンの詳細な例は図11を参照しながら説明する。ステップ1070で、統計収集タイマが所望の統計収集間隔の長さのリセットされる。

【0054】図11に再割り当てルーチン(ステップ1060)の例を示す。ステップ1110で、TOは負荷上限値(TH)を超えたサーバ・セットを含む。ステップ1115で、kをTOで最も負荷が高いサーバのインデックスとする。ステップ1120で、TUは負荷上限値を超えなかったサーバのセットを含む。ステップ1125で、lをTUで最も負荷が低いサーバのインデックスとする。ステップ1130でiをクラス負荷 $CA(i)$ が最小のサーバkに割り当てられたクラスとする。ステップ1135で、サーバlへのクラスiの再割り当てによってサーバlの負荷が上限値を超えない場合、すなわち、 $SA(l) + CA(i) \leq TH$ の場合、クラスiはステップ1140でサーバkからサーバlに再割り当てされる( $C(i)$ をlに変更して) $SA(l)$ 及び $SA(k)$ 。ステップ1145で、 $SA(l)$ と $SA(k)$ は更新されてクラス再割り当てを反映する。特に、 $SA(l)$ は $CA(i)$ によってインクリメントされ、 $SA(k)$ は $CA(i)$ によってデクリメントされる。そうでない場合、ステップ1160でサ

サーバは過負荷のサーバから負荷を受け入れることができないためTUから削除される。ステップ1150で、サーバkの負荷が上限値をまだ超える場合、すなわち、 $SA(k) > TH$ の場合、ステップ1130が再実行される。そうでない場合、ステップ1155でサーバkは負荷が上限値をもはや超えていないため、TOから削除される。ステップ1170で、TOが空きでない場合、ステップ1115が再実行される。ステップ1165で、TUが空きでない場合、ステップ1125が再実行される。

【0055】当業者は本発明の精神と範囲を逸脱せずに本発明の別の実施形態と拡張形態を採用できることを容易に理解するであろう。例えばステップ1140で、再割り当ては単一のクラス(図3)をサーバkからサーバiへ移動して負荷の不均衡を軽減する簡易型の貪欲な方法である。この拡張形態では、負荷平衡を向上させられる場合にはサーバkの単一のクラスをサーバiの別のクラスと交換することができる。ステップ1135で、サーバiが負荷上限値を超えていない場合に再割り当てが発生する。この判定基準は全体の過負荷が軽減されるかどうかという基準に緩和することができる。さらに、クラス負荷のいずれか $CA(i)$ がTHを超える場合、各サーバがそのクラスの要求のほんの一部を受け付ける複数のサーバにそのクラス負荷を割り当てることができる。アービトラータはサーバに割り当てられた負荷の分担部分に従って、見込みでサーバをそのクラスのリクエストに割り当てることができる。同様の再割り当てがサーバ(208)で実施できる。

【0056】さらに、好ましい実施形態ではクラスタ内の全サーバは同じ処理能力があるものとされる。当業者はこれが異種サーバを含むように容易に拡張できることを容易に理解するであろう。異種サーバの場合は、負荷平衡は正規化して受信した要求数を処理能力で割った値になる。特に、 $SA(j)$ はサーバjの処理能力によって正規化できる。

【0057】図11はクラスサーバ割り当てを動的に次第に改良する方法の例を示すことに注意されたい。当業者はまた、本発明の精神と範囲を逸脱せずに元のクラスサーバ割り当てテーブルを提供する別の多くの方法を用いることができることを理解するであろう。事前の作業負荷情報が入手できない場合、ランダムまたはラウンド・ロビン・クラスサーバ割り当てが使用できる。情報が得られる場合、最小処理時間優先(LPT)アルゴリズムを使用できる。各クラスはそのアクセス負荷の降順にソートされる。リストの最上段にあるクラス(すなわち、最も負荷が高いクラス)が最初にリストから外されて現在割り当てられている最も負荷が低いサーバに割り当てられる。そのサーバの割り当てられた負荷はこれに従って調整される。この処理がすべてのクラスが割り当てられるまで繰り返される。

【0058】図12にマッピング要求ハンドラ(245)の例を示す。図示のように、ステップ1210で、オブジェクトid(例:URL)が例えば従来のハッシングその他の方法でそのクラスにマッピングされる。この方法は例えば論理的にURL論理の最初の4バイトとURLの最後の4バイトの論理和をとり、得られた数字をハッシュ・テーブル・サイズで除算することで実行できる。余りの数は0とハッシュ・テーブル・サイズ、マイナス1の間の数字になる。この余りの数はハッシュ・テーブルへのインデックスを表す。ステップ1220で、クラスサーバ・マッピングが割り当てテーブル(225)から決定される。ステップ1230で、マッピング情報がリクエスタへ送信される。

【0059】図13にリクエスタ(203)論理の例を示す。ステップ1310で、リクエスタは入力待ちとなる。ステップ1315で、オブジェクト要求に関してステップ1320でオブジェクト要求生成ルーチンが起動される。オブジェクト要求生成ルーチンはサーバ・キャッシュでのローカリティを向上させるためにオブジェクト識別子(例:URL)に基づいてどのサーバ(IP)アドレスを選択するか決定する。オブジェクト要求生成ルーチンの詳細は図14を参照しながら説明する。ステップ1315で、受信した入力オブジェクト要求でない場合、処理はステップ1350へ進む。ステップ1350で、(以前に要求された)オブジェクトが返送された場合、ステップ1360で、オブジェクト(HTTP)ヘッダに再割り当てラベル(Rラベル)が含まれているかチェックされる。含まれている場合、ステップ1365で、ローカル割り当てテーブル(270)が更新されてクラスサーバ割り当ての変更を反映する。ステップ1370で、受信したオブジェクトが処理される。ステップ1325で、(ステップ1440で以前に要求された)マッピング要求が返送されると、ステップ1330で、(ペンディングの)オブジェクト要求が指定されたサーバへ送信される。ステップ1340で、ローカル割り当てテーブル(270)が更新されてマッピングする要求に基づくクラスサーバ再割り当ての変更を反映する。ステップ1335で、本発明の対象ではない他のタイプの入力(プッシュ・オブジェクトなど)に関して、該当する各種ハンドラを起動できる。

【0060】図14にオブジェクト要求生成(267)論理の例を示す。図示のように、ステップ1410で、オブジェクトはそのオブジェクト・クラスにマッピングされる。ステップ1420で、該当するサーバがクラスサーバ割り当てテーブルにない場合、ステップ1440でマッピング要求がアービトラータへ送信される(オブジェクト要求はこうして図13のステップ1330に示すようにマッピング要求が完了するまで表示される)。該当するサーバがクラスサーバ割り当てテーブルにある場合、ステップ1430で、オブジェクト要求

はクラスサーバ割り当てテーブルが指定するサーバへ送信される。

【0061】当業者は本発明の精神と範囲を逸脱することなく本発明のさまざまな拡張形態が使用できることを容易に理解するであろう。例えば、オブジェクト要求生成ルーチン（ステップ1440）で、マッピング要求を発行する代わりにクラス内の任意のサーバが選択できる。マッピング要求テーブルはそれぞれのクラスサーバ・マッピングの有効な間隔を含むこともできる。この間隔が満了すると、そのクラスの次のオブジェクト要求にตอบสนองしてマッピング要求を発行する（ステップ1440で）ことができる。

【0062】当業者はまた本発明がサーバへの階層的なオブジェクト識別子のマッピングに適用できることを理解するであろう。例えば、図15に示すように、本発明は従来のドメイン・ネーム・サーバ（DNS）またはTCPベースのルーティングとの連携動作が可能である。ここで、クラスサーバ割り当てテーブルは好ましくは各クラス（図3）を仮想サーバに割り当てる。仮想サーバの数はサーバ・クラス内のサーバの実際の数より多い。次にDNS（167）及びTCPルータは各仮想サーバをクラス内の実サーバの1つに動的にマッピングする。

【0063】さらに、メタ情報を要求されたオブジェクトに便乗させることでサーバ位置に関するルーティング情報を更新するという概念を使ってインターネット内の従来のDNSルーティングを更新できる。この処理は、オブジェクトのURL（またはクラス）を使ってサーバ割り当てを実行してキャッシュ・ヒット率を向上させる機能とは無関係である。DNSルーティングは複製の情報を備えた複数のウェブ・サーバ間で負荷平衡を試みるだけである（例として、コラジャンニ（Colajanni）、M.、ユー（Yu）、P.、ディアス（Dias）、D.、の「Scheduling Algorithms for Distributed Web Servers」、IBM Research Report、RC20680、1997年1月を参照）。従来のDNSは各ネームアドレス・マッピングのTTL間隔を備える。このマッピングは各種ネーム・サーバでキャッシングされる。この結果、DNSはクラスタの負荷平衡に使用された場合には制御範囲が限定される。本発明によれば、クラスタ内のサーバが過負荷になると、代替サーバIPアドレスが返送されるオブジェクトに（好ましくはPICSラベルまたは同等の機構を使って）「便乗でき」（ネットワーク・トラフィックを追加せずに）、トラフィック・フローをクラスタ内の他のサーバへ転送して負荷平衡を向上させることができる。

【0064】好ましい実施形態では、DNS（167）は各リクエストから発行される要求の数を収集し、リク

エスターサーバ割り当てテーブルを生成してサーバ間の負荷の平衡を保つ。（異種サーバの場合、割り当てられた負荷をサーバの処理能力に比例させることができる。）（ネームアドレス）マッピング要求がDNS（167）に到着すると、サーバ（161...163）が割り当てテーブルのリクエスト名（またはIPアドレス）に基づいて割り当てられる。マッピングは階層構造でマルチレベル、例えば、URL=>クラス=>仮想サーバ=>サーバという図式になる。DNS（167）は負荷統計を収集し、TTLよりも（はるかに）短い測定間隔に基づいて割り当てテーブル（225）を更新することができる。このように、新しい割り当てテーブルを迅速に生成して負荷条件をより正確に反映させられる。すべてのサーバ（161...163）はDNS（167）から最新バージョンの割り当てテーブル（225）を入手する。前記と同様に、リクエスト（110...153）はこの変更を通知される必要がない。リクエストは以前の（ネームアドレス）マッピングに基づいて要求を送信できる。ただし、サーバがそのサーバに割り当てられていないリクエストから要求を受信した場合、リクエストに将来要求を発行する先のサーバ（161...163）を通知する。現在の要求はそのまま処理され、新しい割り当て情報はPICSその他の機構を使って、応答または返送されるオブジェクトに便乗させられる。サーバが過負荷になった場合、DNS（167）へアラーム信号を送信できる。アラームを受信するたびに、DNS（167）は割り当てテーブルを再計算してすべての過負荷状態のサーバに割り当てられた要求の数を軽減できる。リクエストは割り当てテーブルがクラスサーバ割り当てになるようにクラスに区画することもできる。

【0065】DNS（167）ルーティング論理の例を図15を参照しながら説明する。リクエスト（110）がDNS（167）によってサーバ（162）が割り当てられているものとする。従来技術では、このマッピングはあるTTL間隔、例えば、5分間のTTL間隔だけ有効である。本発明によれば、更新された割り当てテーブルは例えば1分間というより短い間隔で生成でき、リクエスト（110）は負荷を減らされたサーバ（163）に割り当てられる。リクエスト（110）はこの時点では変更を知る必要がない。リクエストは同じサーバ（162）へ次の要求を送信する。ただし、サーバ（162）はDNS（167）から新しい割り当てテーブルを受信している。サーバ（162）は要求を処理するが、リクエスト（110）へ返送するオブジェクトにメッセージを便乗させてサーバ（163）へ将来の要求を送信する。これでトラフィックを追加せずにTTLの悪影響を除去できる。

【0066】当業者は本発明の動的ルーティング方法が、リクエストの一部がルーティング・プロトコルを認



識せずキャッシュ・ヒット率及び負荷平衡を向上させる協動に参加しない既存のプロキシ／クライアント・ステーションである異種リクエスト環境でも有効であるということとをさらに理解するであろう。

【0067】まとめとして、本発明の構成に関して以下の事項を開示する。

【0068】(1) オブジェクト要求を処理するサーバ・ノードの集合が用いる動的ルーティング方法であって、メタ情報を要求されたオブジェクトに便乗させるステップと、メタ情報に従ってサーバに対するルーティング情報を動的に更新するステップを含む方法。

(2) サーバ・ノード間で負荷を平衡させるステップと、要求されたオブジェクトのキャッシュ・ヒット率を最適化するステップをさらに含む上記(1)に記載の方法。

(3) サーバ・ノード間で負荷を平衡させるステップをさらに含む上記(1)に記載の方法。

(4) サーバの集合がインターネット環境でのプロキシ・サーバのクラスタまたはウェブ・サーバのクラスタを含む上記(1)に記載の方法。

(5) オブジェクト要求を処理する複数のプロキシ・サーバ・ノード間の動的ルーティング方法であって、要求されたオブジェクトのオブジェクト識別子に従ってサーバを割り当てるステップと、前記割り当てステップにตอบสนองして、更新されたサーバ割り当てをオブジェクト・リクエストに通知するステップを含む方法。

(6) オブジェクト要求を処理する複数のウェブ・サーバ・ノード間の動的ルーティング方法であって、同じホスト名またはアドレスに向けられたオブジェクト要求を、要求されたオブジェクトの識別子に従ってクラスタ内の異なるサーバに割り当てるステップと、更新されたサーバ割り当てを、オブジェクト・リクエストに通知するステップと、オブジェクト・リクエストが後続のオブジェクト要求に対する要求されたオブジェクトの更新されたサーバ割り当てを動的に保守するステップを含む方法。

(7) オブジェクト識別子の階層的マッピングに従ってサーバを割り当てるステップをさらに含む上記(1)に記載の方法。

(8) オブジェクト識別子がURLである上記(7)に記載の方法。

(9) オブジェクト識別子をクラスにマッピングするステップと、クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む上記(2)に記載の方法。

(10) 前記マッピング・ステップがオブジェクト識別子をハッシュ・テーブル経由でクラスまたはハッシュ・クラスにマッピングするステップをさらに含む上記(9)に記載の方法。

(11) 各オブジェクト要求のオブジェクト識別子をク

ラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、前記マッピング・ステップにตอบสนองしてクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記(2)に記載の方法。

(12) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記(2)に記載の方法。

(13) 前記負荷平衡ステップが各クラスに関連付けられた負荷の関数としてのクラスをサーバ・ノードに割り当てるステップをさらに含む上記(2)に記載の方法。

(14) 前記負荷平衡ステップがクラスをサーバに増加方向に再び割り当てるステップをさらに含む上記(2)に記載の方法。

(15) 前記便乗ステップがPICSプロトコルの使用を含みルーティング情報を更新する上記(1)に記載の方法。

(16) 各要求元ノードがオブジェクトの現在のサーバ割り当て要求を送信するステップをさらに含む上記(1)に記載の方法。

(17) 前記要求送信ステップがPICSプロトコルを使って要求されたオブジェクトのクラスに基づいて現在のサーバ割り当てを決定する上記(16)に記載の方法。

(18) オブジェクト識別子の階層マッピングに従ってサーバを割り当てるステップが、各クラスを仮想サーバ・ノードに割り当てるステップと、仮想サーバ・ノードを実際のサーバ・ノードの動的にマッピングするステップをさらに含む上記(7)に記載の方法。

(19) サーバ・ノードの集合が、前記の動的マッピング・ステップがネームアドレス・マッピング及びネームアドレス・マッピングに関連付けられたタイムアウト(TTL)間隔を含み、DNSがTTLより短い間隔で実際のサーバ・ノードに仮想サーバ・ノードを動的にマッピングするステップと、更新されたサーバのマッピングをすべてのサーバに通知するステップをさらに含み、前記メタ情報が更新されたサーバのマッピングを含み、前記の動的にルーティング要求を更新するステップが更新されたサーバ・マッピングに従って後続のオブジ

ェクト要求をルーティングするステップを含むドメイン・ネーム・サーバ (DNS) を含む上記 (18) に記載の方法。

(20) サーバ・ノードの集合がTCPルータを含み、ルータが実際のプロキシ・ノードに仮想サーバ・ノードを動的にマッピングするステップをさらに含む上記 (18) に記載の方法。

(21) すべてのリクエストが前記の動的更新ステップを実行するように適合されていない異種リクエスト環境をさらに含む上記 (1) に記載の方法。

(22) 前記サーバ割り当てステップがオブジェクト識別子の階層マッピングに従ってサーバを割り当てるステップを含む上記 (5) に記載の方法。

(23) オブジェクト識別子をクラスにマッピングするステップと、クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む上記 (5) に記載の方法。

(24) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、前記マッピング・ステップにตอบสนองしてクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記 (5) に記載の方法。

(25) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記 (5) に記載の方法。

(26) 前記オブジェクト要求割り当てステップがオブジェクト識別子の階層マッピングに従ってオブジェクト要求を割り当てるステップを含む上記 (6) に記載の方法。

(27) オブジェクト識別子をクラスにマッピングするステップと、クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む上記 (6) に記載の方法。

(28) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、前記マッピング・ステップにตอบสนองしてクラスサーバ割り

当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記 (6) に記載の方法。

(29) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記 (6) に記載の方法。

(30) サーバ・ノードの集合への要求をクラス内の異なるサーバに割り当てることができるサーバ・ノードの集合の動的なルーティング方法であって、リクエストがリクエスト識別子またはIPアドレスのいずれかを含むマッピング要求を定期的にサーバに送信するステップと、前記のリクエスト識別子またはIPアドレスのいずれかをリクエスト負荷及びサーバ容量のいずれかに基づいてサーバ・ノードの集合内のサーバにマッピングするステップと、前記マッピング・ステップにตอบสนองしてサーバ・マッピングをすべてのサーバに送信するステップと、サーバのいずれかがそのサーバに割り当てられていないリクエストから要求を受信した場合に、サーバがリクエストにリクエストサーバ割り当ての変更を通知するステップを含む方法。

(31) リクエストに通知する前記ステップが、サーバが要求を処理するステップをさらに含む上記 (30) に記載の方法。

(32) リクエスト識別子またはIPアドレスをクラスに区画化するステップと、アービトラータ・サーバとサーバの集合にあるクラスサーバ割り当てテーブルを保守するステップをさらに含む上記 (30) に記載の方法。

(33) アービトラータ・サーバがインターネット環境のDNSを含む上記 (32) に記載の方法。

(34) オブジェクト識別子またはIPアドレスの階層マッピングに従ってサーバを割り当てるステップをさらに含む上記 (3) に記載の方法。

(35) リクエスト識別子をクラスにマッピングし、クラスに基づいてサーバを割り当てるステップをさらに含む上記 (3) に記載の方法。

(36) サーバが更新されたメタ情報をリクエストへ送信し、リクエストが割り当てを更新するステップをさらに含む上記 (3) に記載の方法。

(37) マシン上で実行され、オブジェクト要求を処理するサーバ・ノードの集合でオブジェクト要求を動的に



ルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、メタ情報を要求されたオブジェクトに便乗させるステップと、メタ情報に従ってサーバ割り当てに対するルーティング情報を動的に更新するステップを含むプログラム記憶装置。

(38) サーバ・ノード間で負荷を平衡させるステップと、要求されたオブジェクトのキャッシュ・ヒット率を最適化するステップをさらに含む上記(37)に記載のプログラム記憶装置。

(39) サーバ・ノード間で負荷を平衡させるステップをさらに含む上記(37)に記載のプログラム記憶装置。

(40) サーバの集合がインターネット環境でのプロキシ・サーバのクラスタまたはウェブ・サーバのクラスタを含む上記(37)に記載のプログラム記憶装置。

(41) マシン上で実行され、オブジェクト要求を処理するサーバ・ノードの集合でオブジェクト要求を動的にルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、要求されたオブジェクトのオブジェクト識別子に従ってサーバを割り当てるステップと、前記割り当てステップにตอบสนองして、更新されたサーバ割り当てをオブジェクト・リクエストに通知するステップを含むプログラム記憶装置。

(42) 前記サーバ割り当てステップがオブジェクト識別子の階層マッピングに従ってサーバを割り当てるステップを含む上記(41)に記載のプログラム記憶装置。

(43) オブジェクト識別子をクラスにマッピングするステップと、クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む上記(41)に記載のプログラム記憶装置。

(44) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、前記マッピング・ステップにตอบสนองしてクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記(41)に記載のプログラム記憶装置。

(45) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減する

リクエストでのサーバ選択方法が提供される上記(41)に記載のプログラム記憶装置。

(46) マシン上で実行され、オブジェクト要求を処理する複数のウェブ・サーバ・ノードでオブジェクト要求を動的にルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、同じホスト名またはアドレスに向けられたオブジェクト要求を、要求されたオブジェクトの識別子に従ってクラスタ内の異なるサーバに割り当てるステップと、更新されたサーバ割り当てを、オブジェクト・リクエストに通知するステップと、オブジェクト・リクエストが後続のオブジェクト要求に対する要求されたオブジェクトの更新されたサーバ割り当てを動的に保守するステップを含むプログラム記憶装置。

(47) 前記オブジェクト要求割り当てステップがオブジェクト識別子の階層マッピングに従ってオブジェクト要求を割り当てるステップを含む上記(46)に記載のプログラム記憶装置。

(48) オブジェクト識別子をクラスにマッピングするステップと、クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む上記(46)に記載のプログラム記憶装置。

(49) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、前記マッピング・ステップにตอบสนองしてクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記(46)に記載のプログラム記憶装置。

(50) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、前記選択ステップにตอบสนองしてクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記(46)に記載のプログラム記憶装置。

(51) オブジェクト識別子の階層マッピングに従ってサーバを割り当てるステップをさらに含む上記(37)に記載のプログラム記憶装置。

(52) オブジェクト識別子がURLである上記(51)に記載のプログラム記憶装置。

(53) オブジェクト識別子の階層マッピングに従ってサーバを割り当てるステップが、各クラスタを仮想サー

バ・ノードに割り当てるステップと、仮想サーバ・ノードを実際のサーバ・ノードに動的にマッピングするステップをさらに含む上記(51)に記載のプログラム記憶装置。

(54) サーバ・ノードの集合が、前記の動的マッピング・ステップがネームアドレス・マッピング及びネームアドレス・マッピングに関連付けられたタイムアウト(TTL)間隔を含み、DNSがTTLより短い間隔で実際のサーバ・ノードに仮想サーバ・ノードを動的にマッピングするステップと、更新されたサーバのマッピングをすべてのサーバに通知するステップをさらに含み、前記メタ情報が更新されたサーバのマッピングを含み、前記の動的なルーティング要求を更新するステップが更新されたサーバ・マッピングに従って後続のオブジェクト要求をルーティングするステップを含むドメイン・ネーム・サーバ(DNS)を含む上記(53)に記載のプログラム記憶装置。

(55) サーバ・ノードの集合がTCPルータを含み、ルータが実際のプロキシ・ノードに仮想サーバ・ノードを動的にマッピングするステップをさらに含む上記(53)に記載のプログラム記憶装置。

(56) オブジェクト識別子をクラスにマッピングするステップと、クラス及びクラスサーバ割り当てテーブルに基づいてサーバを割り当てるステップをさらに含む上記(38)に記載のプログラム記憶装置。

(57) 前記マッピング・ステップがオブジェクト識別子をハッシュ・テーブル経由でクラスまたはハッシュ・クラスにマッピングするステップをさらに含む上記(56)に記載のプログラム記憶装置。

(58) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にアービトラータにマッピング要求を発行するステップと、前記マッピング・ステップに回答してクラスサーバ割り当てテーブルを更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記(38)に記載のプログラム記憶装置。

(59) 各オブジェクト要求のオブジェクト識別子をクラスにマッピングするステップと、クラスサーバ割り当てテーブルに有効なサーバ割り当てがない場合にサーバを選択するステップと、前記選択ステップに回答してクラスサーバ割り当てテーブルを選択したサーバで更新するステップを含む、サーバ選択のためのリクエスト・ノードでクラスサーバ割り当てテーブルを保守するステップを含む、割り当て要求トラフィックを軽減するリクエストでのサーバ選択方法が提供される上記(38)に記載のプログラム記憶装置。

(60) 前記負荷平衡ステップが各クラスに関連付けら

れた負荷の関数としてのクラスをサーバ・ノードに割り当てるステップをさらに含む上記(38)に記載のプログラム記憶装置。

(61) 前記負荷平衡ステップがクラスをサーバに増加方向に再び割り当てるステップをさらに含む上記(38)に記載のプログラム記憶装置。

(62) 前記便乗ステップがPICSプロトコルの使用を含みルーティング情報を更新する上記(37)に記載のプログラム記憶装置。

(63) 各要求元ノードがオブジェクトの現在のサーバ割り当て要求を送信するステップをさらに含む上記(37)に記載のプログラム記憶装置。

(64) 前記要求送信ステップがPICSプロトコルを使って要求されたオブジェクトのクラスに基づいて現在のサーバ割り当てを決定する上記(63)に記載のプログラム記憶装置。

(65) すべてのリクエストが前記の動的更新ステップを実行するように適合されていない異種リクエスト環境をさらに含む上記(37)に記載のプログラム記憶装置。

(66) オブジェクト識別子またはIPアドレスの階層マッピングに従ってサーバを割り当てるステップをさらに含む上記(39)に記載のプログラム記憶装置。

(67) リクエスト識別子をクラスにマッピングし、クラスに基づいてサーバを割り当てるステップをさらに含む上記(39)に記載のプログラム記憶装置。

(68) サーバが更新されたメタ情報をリクエストへ送信し、リクエストが割り当てを更新するステップをさらに含む上記(39)に記載のプログラム記憶装置。

(69) マシン上で実行され、オブジェクト要求を処理するサーバ・ノードの集合でオブジェクト要求を動的にルーティングする方法ステップを実行する命令プログラムを明らかに実施したマシン可読プログラム記憶装置であって、前記方法ステップが、リクエストがリクエスト識別子またはIPアドレスのいずれかを含有するマッピング要求を定期的にサーバに送信するステップと、前記のリクエスト識別子またはIPアドレスのいずれかをリクエスト負荷及びサーバ容量のいずれかに基づいてサーバ・ノードの集合内のサーバにマッピングするステップと、前記マッピング・ステップに回答してサーバ・マッピングをすべてのサーバに送信するステップと、サーバのいずれかがそのサーバに割り当てられていないリクエストから要求を受信した場合に、サーバがリクエストにリクエストサーバ割り当ての変更を通知するステップを含むプログラム記憶装置。

(70) リクエストに通知する前記ステップが、サーバが要求を処理するステップをさらに含む上記(69)に記載のプログラム記憶装置。

(71) リクエスト識別子またはIPアドレスをクラスに区画化するステップと、アービトラータ・サーバとサ

サーバの集合にあるクラスサーバ割り当てテーブルを保守するステップをさらに含む上記(69)に記載のプログラム記憶装置。

【72】アービトラータ・サーバがインターネット環境のDNSを含む上記(71)に記載のプログラム記憶装置。

【図面の簡単な説明】

【図1】本発明に適用できるインターネット環境を示す図である。

【図2】本発明の機能を備える一般環境の例を示す詳細図である。

【図3】「クラスサーバ」割り当てテーブルの例を示す図である。

【図4】図2のサーバ論理の例を示す図である。

【図5】サーバのオブジェクト・ハンドラの例を示す図である。

【図6】サーバのオブジェクト・ハンドラの動的割り当てルーチンの例を示す図である。

【図7】サーバのオブジェクト要求ハンドラの例を示す図である。

【図8】サーバの統計報告ルーチンの例を示す図である。

【図9】図2のアービトラータ論理の例を示す図である。

【図10】アービトラータの統計及び評価ルーチンの例を示す図である。

【図11】アービトラータの統計及び評価ルーチンの再割り当てルーチンの例を示す図である。

【図12】アービトラータのマッピング要求ハンドラの例を示す図である。

【図13】図2のリクエスト論理の例を示す図である。

【図14】リクエスト論理のオブジェクト要求の生成例を示す図である。

【図15】ドメイン・ネーム・サーバ(DNS)を含む図1のサーバ・クラスタの例を示す図である。

【符号の説明】

103 サーバ・クラスタ

105 ネットワーク

110 リクエスト1

112 PC

118 リクエスト・ノードK

120 PC

125 プロキシ・サーバ・ノード

126 プロキシ・サーバ

127 プロキシ・サーバ

150 PC

151 PC

152 PC

153 PC

161 サーバ・ノード

162 サーバ・ノード

163 サーバ・ノード

201 ネットワーク

202 リクエスト・ノード1

203 リクエスト・ノードK

205 サーバ・クラスタ

206 サーバ1

208 サーバM

210 メモリ

212 オブジェクト要求ハンドラ

214 オブジェクト・ハンドラ

218 統計報告ルーチン

220 キャッシュ・マネージャ

225 割り当てテーブル

227 CPU

230 記憶装置

235 アービトラータ

240 CPU

242 プログラム記憶

245 メモリ

248 マッピング要求ハンドラ

250 統計及び評価

260 CPU

263 メモリ

265 記憶装置

267 オブジェクト要求の生成ルーチン

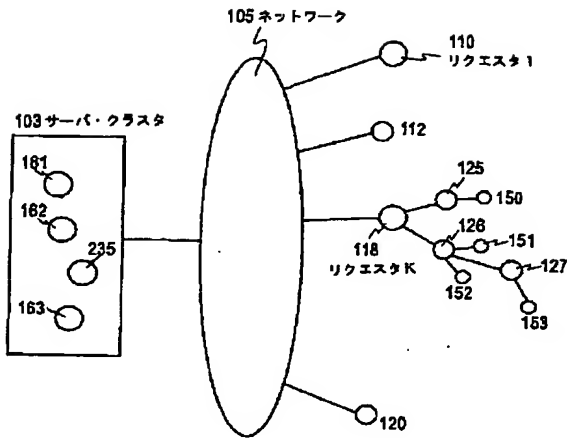
270 割り当てテーブル

【図3】

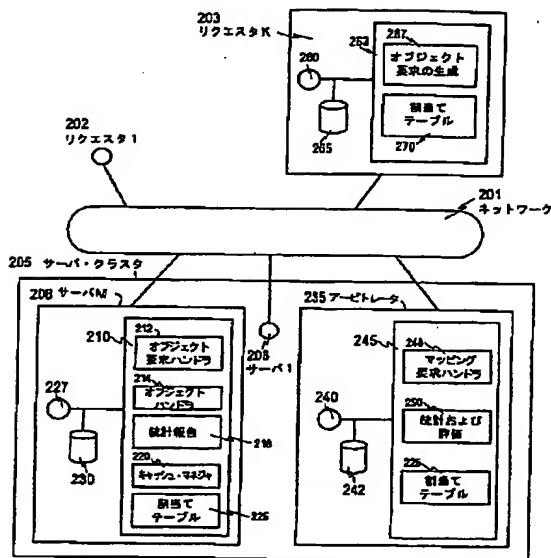
割り当てテーブル  
N=16, M=3

クラス	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
サーバ	2	3	2	2	2	2	2	3	1	3	1	2	2	3	2	3

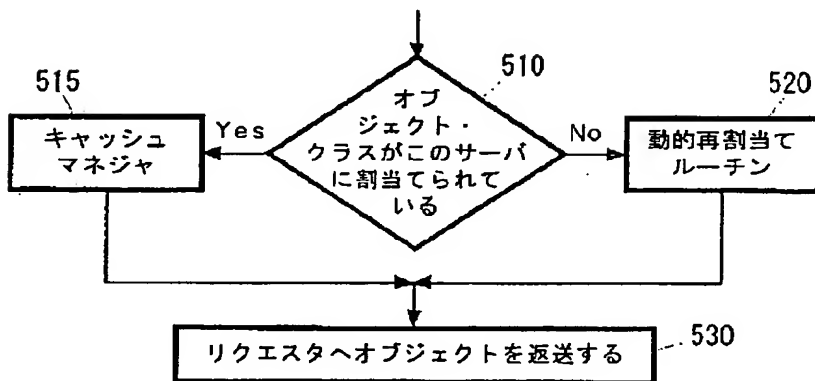
【図 1】



【図 2】

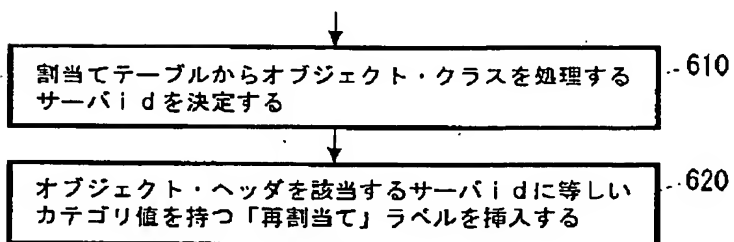


【図 5】



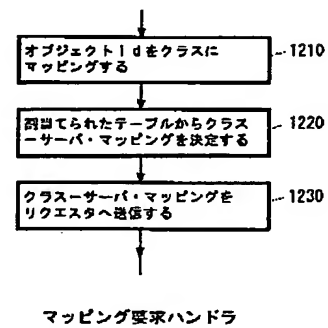
オブジェクト・ハンドラ

【図 6】



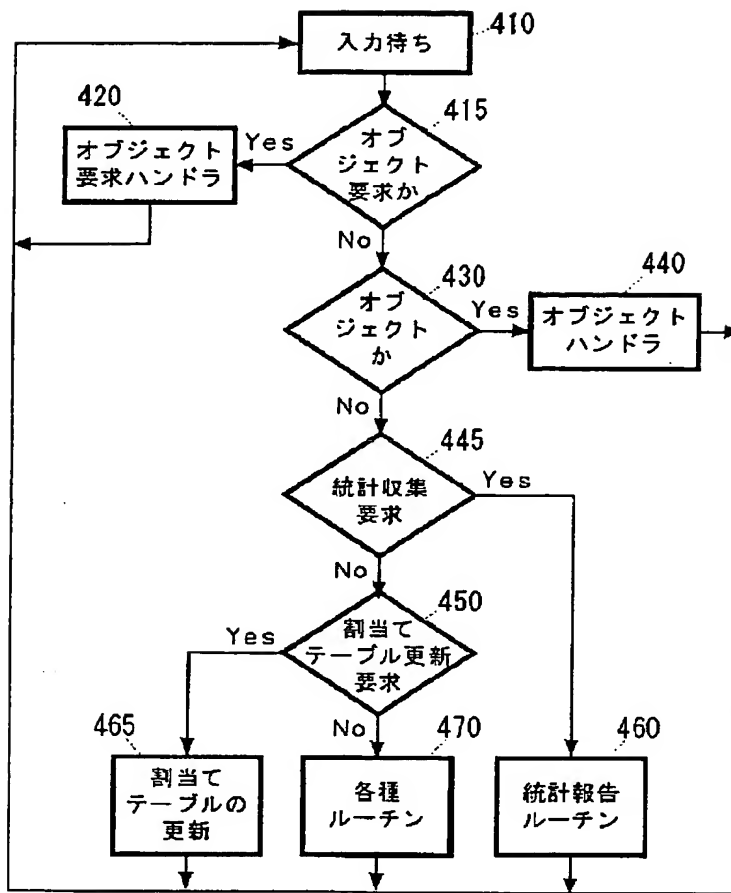
動的再割当てルーチン

【図 12】

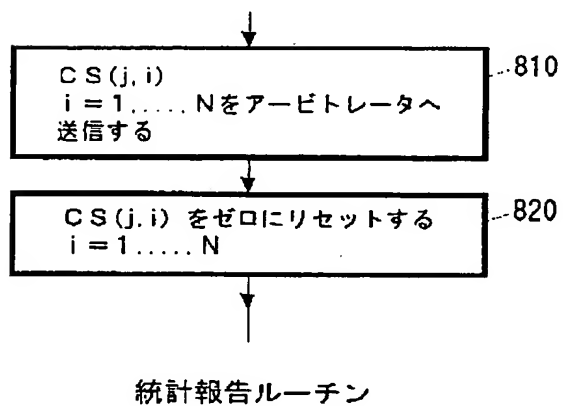


マッピング要求ハンドラ

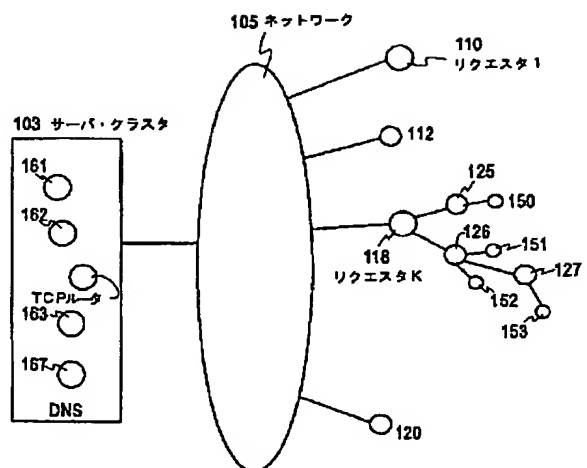
【図 4】



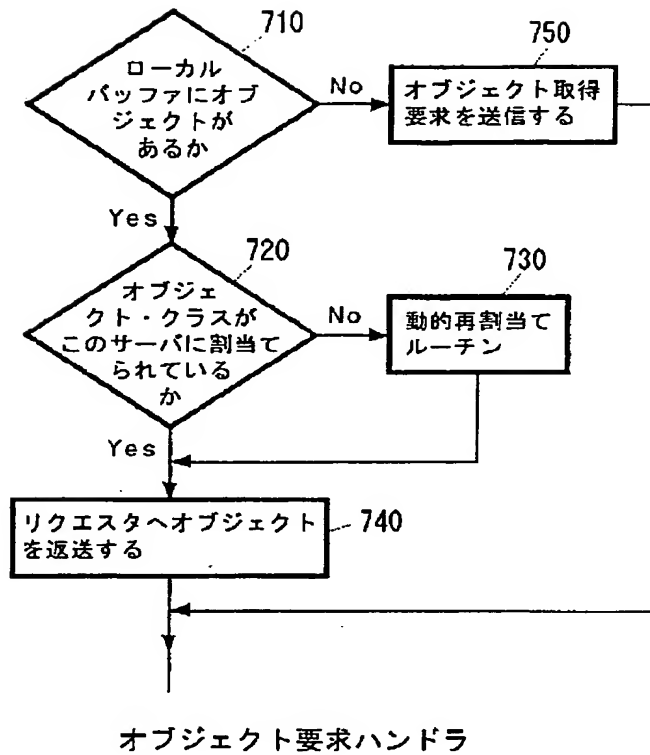
【図 8】



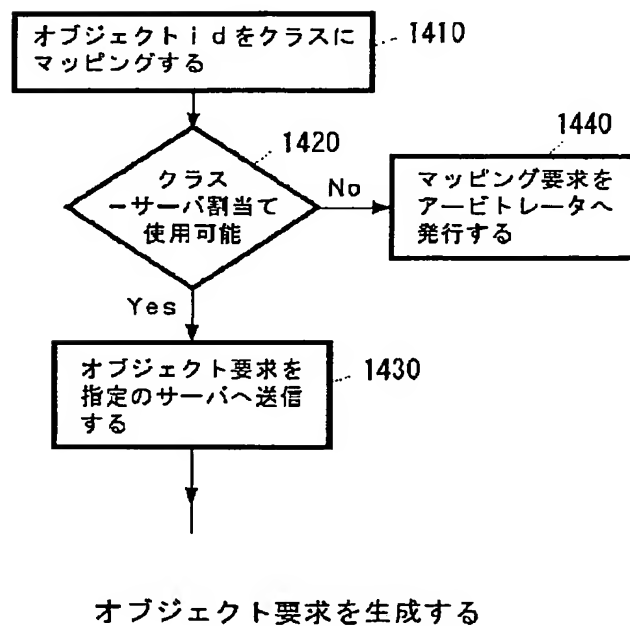
【図 15】



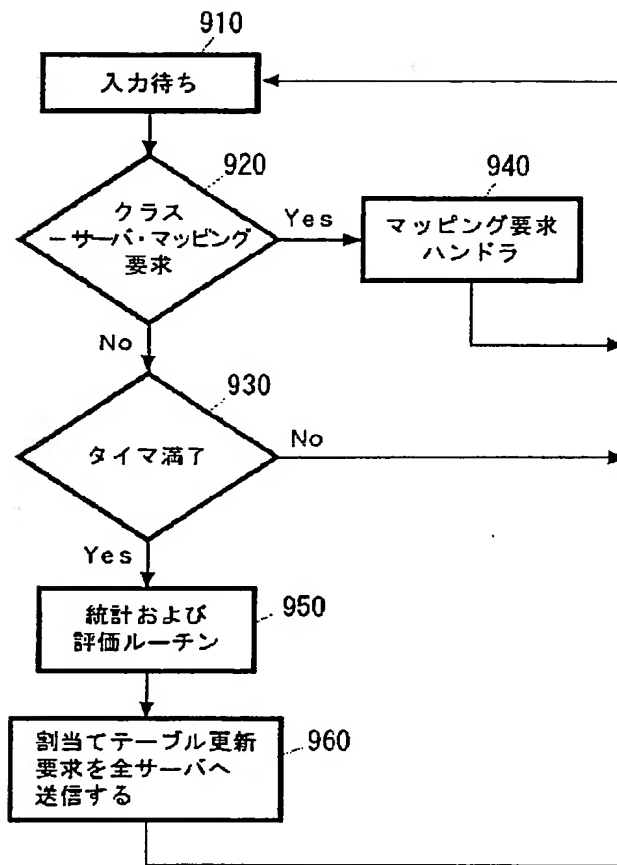
【図 7】



【図 14】

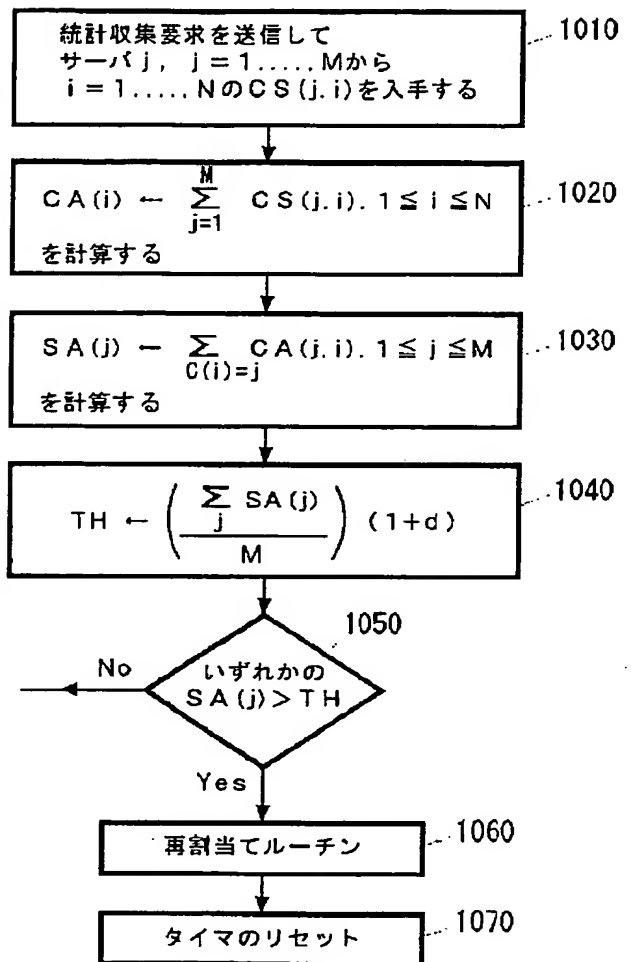


【図 9】



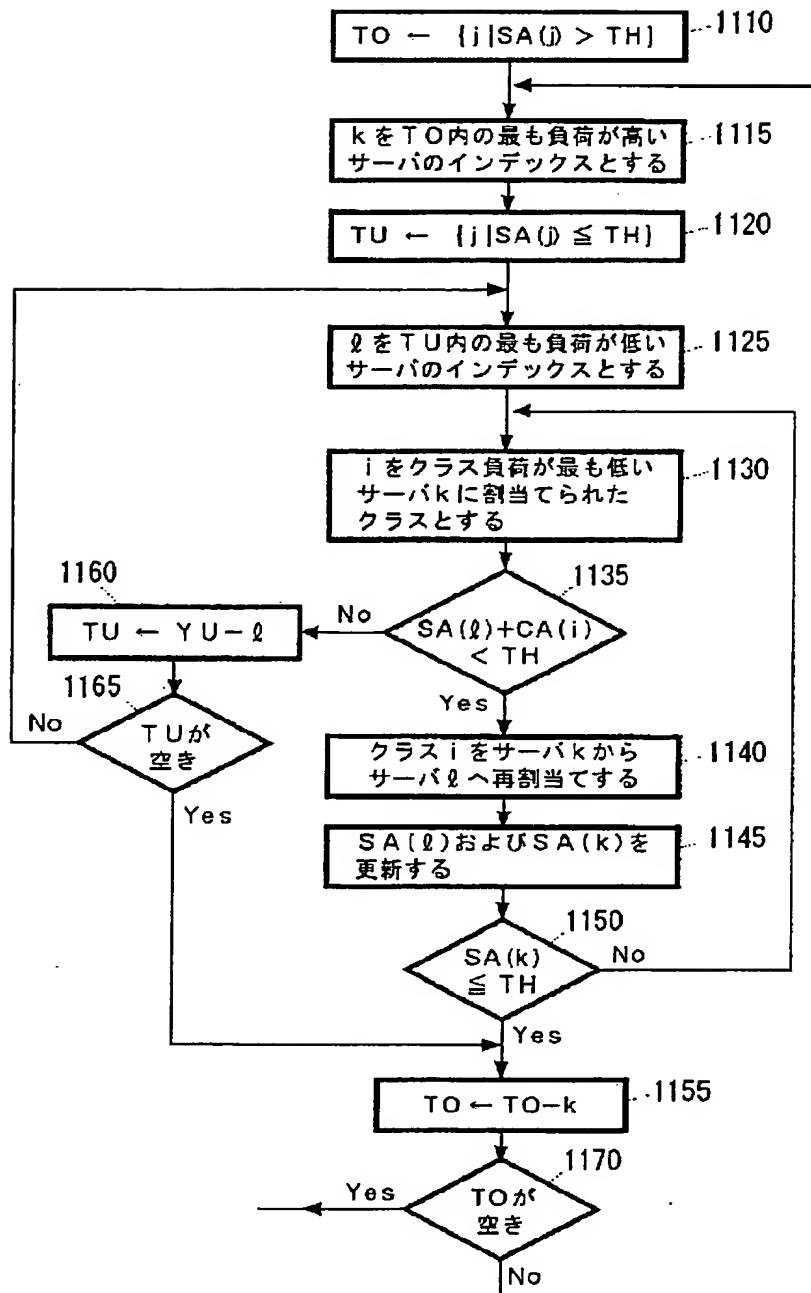
アービトレータ

【図 10】



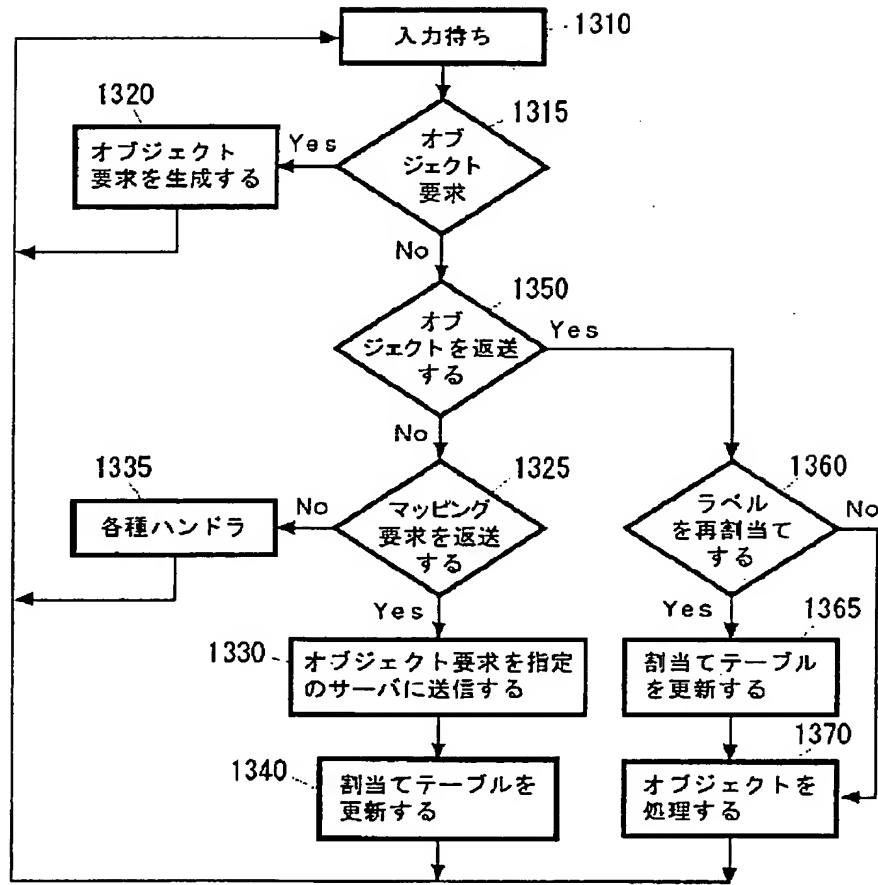
統計および評価ルーチン

【図 11】





【図 13】



リクエスト